Számítógép nélküli feladatok

1. feladat: Program (40 pont)

Egy programban összekeveredtek az értékadó utasítások, lehetnek benne olyan változók, amelyeket a kiszámításuk előtt használunk fel. Sajnos lehetnek hibás programok is, amelyek értékadó utasításait semmilyen sorrendbe nem tudjuk átrendezni úgy, hogy minden értékadás jobboldalán szereplő változó ki legyen számolva, mire az értékadást végre kell hajtani.

Megjegyzés: a képletek tartalma lényegtelen, csak azt kell figyelni, hogy mely változók szerepelnek bennük!

Például az a:=b; b:=c/2+d; c:=a; d:=c/2 értékadások semmilyen sorrendben nem lesznek jók, mert mindig lesz olyan változó, ami még nem használható fel. Ha azonban az a:=b utasítást Beolvas(a)-ra cserélnénk, akkor ez a sorrend jó: Beolvas(a); c:=a; d:=c/2; b:=c/2+d.

A. Az alábbi összekeveredett programokból melyek rendezhetők át úgy, hogy minden változó kapjon értéket a felhasználása előtt?

B. Amelyek nem, azokban adj meg minimális számú értékadást, amelyeket beolvasással helyettesítve az ilyen programok már átrendezhetők lesznek!

C. Adj meg minden programra (a B feladatban átrendezettekre is) egy lehetséges utasítássorrendet!

1. program: a:=4\*b; b:=4\*(c/3+d+e); c:=a/5; d:=3\*c; e:=6; f:=c/3

2. program: a:=4\*b+3\*d+c; b:=3+d; c:=5; d:=3\*c

3. program: a:=4\*b+3\*d; b:=3+d; c:=a-e; d:=3\*c; e:=d\*d

2. feladat: Programstruktúrák kezdete és vége (36 pont)

Egy kifejezésben gömbölyű és szögletes zárójelek lehetnek, tetszőleges mélységben zárójelezve. Készítettünk egy algoritmust, amely ellenőrzi a zárójelek helyes párosítását. A megoldásban az N elemű X vektor tartalmazza a kifejezés karaktereit.

Ellenőr:
 A:=0; B:=0
 Ciklus I=1-től N-ig
 Elágazás
 X(I)="(" esetén A:=A+1; V(A):=B
 X(I)="[" esetén B:=B+1
 X(I)=")" esetén A:=A-1; Ha A<0 akkor HIBA1
 Ha V(A+1)<B akkor HIBA2
 Ha V(A+1)>B akkor HIBA3
 X(I)="]" esetén B:=B-1; Ha B<0 akkor HIBA4
 Elágazás vége
 Ciklus vége
 Ha A>0 akkor HIBA5
 Ha B>0 akkor HIBA6
Eljárás vége.

Milyen hibajelenségek váltják ki a HIBA1..HIBA6 hibajelző utasításokat?

3. feladat: Mit csinál (34 pont)

Az alábbi algoritmus egy N elemű 1-től N-ig indexelt X vektort dolgoz fel, eredményét az U, V, Y változókba írja.

Valami(N,X,Y,U,V):
 T(0):=0
 Ciklus i=1-től N-ig
 T(i):=T(i-1)+X(i)
 Ciklus vége
 Y:=0; U:=0; V:=0 {\*}
 Ciklus a=1-től N-ig
 Ciklus b=a-tól N-ig
 Ha T(b)-T(a-1)>Y akkor Y:=T(b)-T(a-1); U:=a; V:=b
 Ciklus vége
 Ciklus vége
Eljárás vége.

A. Mi lesz a T vektorban a {\*}-gal jelölt helyen, ha N=5, X=(3,-4,4,-3,4)?

B. Mi lesz U,V,Y értéke, ha N=5, X=(3,-4,4,-3,4)?

C. Fogalmazd meg általánosan, hogyan függ T értéke az X vektorban szereplő értékektől!

D. Mi a feltétele annak, hogy U és V értéke 0 maradjon?

E. Fogalmazd meg általánosan, hogyan függ U, V és Y értéke az X vektorban szereplő értékektől (ha U és V nem marad 0)!

4. feladat: Gyöngyök (40 pont)

Egy urnában piros és zöld gyöngyök vannak. Lépésenként véletlenszerűen húzunk két gyöngyöt. Ha mindkettő piros, akkor kivehetünk még egy pirosat (ha nincs harmadik piros, akkor a kettőt is vissza kell tennünk). Ha mindkettő zöld, eltehetjük őket. Ha különböző színűek, akkor vissza kell tennünk őket.

A. Milyen esetekben nem tudunk kivenni gyöngyöt az urnából?

B. Mi a feltétele annak az urnában kezdetben levő gyöngyökre, hogy az urnában a végén ne maradjon piros gyöngy?

C. Mi a feltétele annak az urnában kezdetben levő gyöngyökre, hogy az urnában a végén ne maradjon zöld gyöngy?

D. Mi a feltétele annak az urnában kezdetben levő gyöngyökre, hogy az urnában a végén egy piros gyöngy maradjon?

E. Mi a feltétele annak az urnában kezdetben levő gyöngyökre, hogy az urnában a végén kettő piros gyöngy maradjon?

F. Mi a feltétele annak az urnában kezdetben levő gyöngyökre, hogy az urnában a végén egy zöld gyöngy maradjon?

Számítógépes feladat – VÁLASZTHATÓ

5. feladat: Kikötők (50 pont)

Ismerjük a Duna mentén N kikötő helyét, a Duna feketeerdei forrásától számított kilométerben, távolság szerint növekvő sorrendben.

Készíts programot, amely megadja:

1. a leghosszabb szakaszt, ahol nincs kikötő;
2. azon szomszédos kikötőhármasok számát, amelyekből a két szélső legfeljebb K kilométerre van egymástól;
3. egy középső kikötőt (középső kikötőnek azt nevezzük, amelynek az első és utolsó kikötőtől mért távolsága különbsége a lehető legkisebb)!

A *standard bemenet* első sorában az kikötők száma (3≤N≤100), a Duna hossza (1≤H≤5000) és a K szám van (1≤K≤100). A következő N sorban egy-egy kikötőnek a Duna forrásától mért távolsága van (1≤Ti≤H).

A *standard kimenetre* három sort kell írni, az elsőbe a leghosszabb szakasz hosszát, ahol nincs kikötő, a másodikba azon szomszédos kikötőhármasok számát, amelyben a két szélső legfeljebb K kilométerre van egymástól, a harmadikba pedig egy középső kikötő sorszámát!

Példa:

Bemenet Kimenet

5 4000 30 3720
50 2
60 4
70
80
3800

Számítógép nélküli feladat – VÁLASZTHATÓ

5. feladat: Földrengések (50 pont)

Japánban nagyon sok földrengés van, ismerjük az elmúlt H év N darab nagy földrengésének idejét, a T(1..N) tömbben. Tudjuk mindegyikről, hogy az idei év előtt hány évvel volt (1 és H közötti egész szám), időrendben, a legrégebbivel kezdődően.

Készítettünk egy programot, amely megadja:

1. az elmúlt H év leghosszabb egymást követő évei számát, amikor nem volt földrengés;
2. azon évek számát, amelyeket közvetlenül követő évben is volt földrengés;
3. a leghosszabb időszak évei számát, amikor minden évben volt nagy földrengés;
4. azon évek sorszámát (a bemenetben kapott értéket), amelyek után kevesebb idő telt el a következő földrengésig, mint amennyi közte és az őt megelőző között volt.

A megoldás sajnos hibás lett, keresd meg a hibákat az alábbi algoritmusban!

Földrengések(N,T,H):
 A:=H-T(1); B:=0; C:=0; K:=0; Db:=0
 Ha T(N)>A akkor A:=T(N)
 Ciklus i=2-től N-ig
 Ha T(i-1)-T(i)+1>A akkor A:=T(i-1)-T(i)-1
 Ha T(i-1)=T(i)+1 akkor B:=A+1
 Ha T(i-1)>T(i)+1 akkor Ha K>C akkor K:=C
 K:=1
 különben K:=K+1
 Ciklus vége
 Ha K>C akkor C:=K
 Ciklus i=2-től N-ig
 Ha T(i)-T(i+1)<T(i+1)-T(i) akkor Db:=Db+1; D(Db):=i
 Ciklus vége
Eljárás vége.